

SIDANG TUGAS AKHIR

PENGUKURAN KUALITAS DAN PERBAIKAN STRUKTUR *CODE*
PERANGKAT LUNAK BERBASIS *OBJECT ORIENTED*
PROGRAMMING MENGGUNAKAN METRIK CHIDAMBER
DAN KEMERER. (STUDI KASUS *SOFTWARE ACCOUNTING*
XYZ)

AULA AYUBI || 5211100111

PEMBIMBING 1: FEBY ARTWODINI MUWTADIROH, S.Kom, M.T.

PEBIMBING 2: AMNA SHIFIA NISAFANI, S.Kom, M.Sc



OUTLINE



PENDAHULUAN || LATAR BELAKANG

Kualitas Software

- Salah satu faktor penting dari suatu perangkat lunak adalah kualitas (Yunis, 2014).

Pengukuran Kualitas

- Memberikan pandangan kepada perusahaan atau organisasi

Pengukuran Kuantitatif

- Engineer akan lebih memahami perangkat lunak dengan melihat code pada software tersebut dan menghasilkan nilai kuantitatif (Wahono, 2006).

Peningkatan Kualitas

- Peningkatan kualitas sebuah software mempunyai peran yang penting dalam praktek rekayasa perangkat lunak (Schneidewind, 2002).

Software Accounting XYZ

- Produk dari sebuah softwarehouse yang akan terus menerus dikembangkan dan diperbaiki.

PENDAHULUAN || RUMUSAN MASALAH

Struktur Code

Bagaimana bentuk struktur kode program pada Software Accounting XYZ

Nilai Metrik C.K

Berapa nilai metrik Metrik Chidamber dan Kemerer berdasarkan kode program Software Accounting XYZ

Rekomendasi

Bagaimana bentuk rekomendasi perbaikan yang diberikan kepada pengembang Software Accounting XYZ

PENDAHULUAN || BATASAN MASALAH

Design
Pattern -
Refactoring

Rekomendasi

ISO 9126-1

Kriteria
kualitas

Eclipse IDE
for Java

Analisa

PENDAHULUAN || TUJUAN PENULISAN

Menemukan bentuk struktur kode program pada Software Accounting XYZ

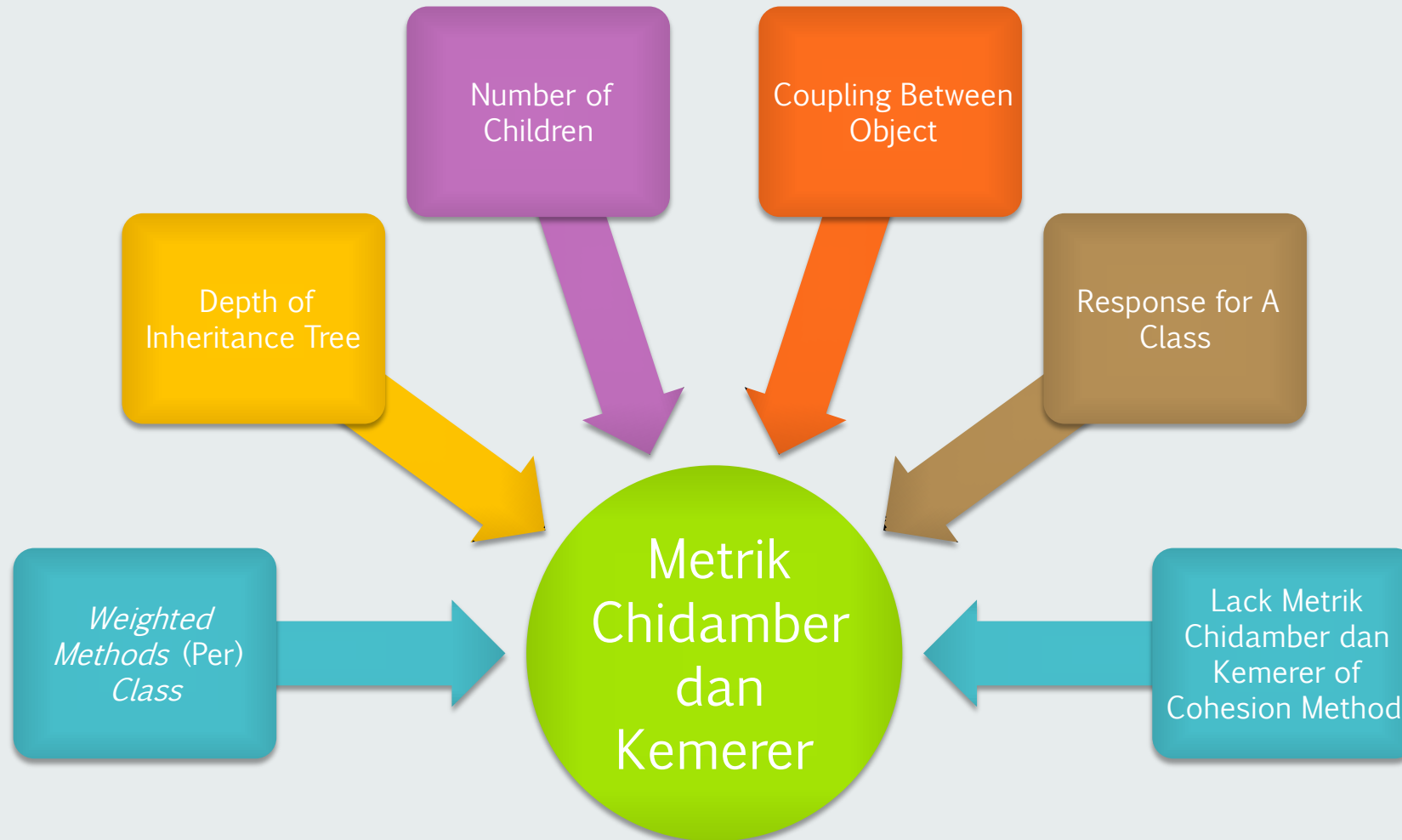
Menghasilkan nilai Metrik Chidamber dan Kemerer yang merepresentasikan kualitas Software Accounting XYZ berdasarkan ISO/IEC 9126-1

Membuat rekomendasi berupa class diagram yang lebih baik berdasarkan Design Pattern

MANFAAT DAN RELEVANSI

- Perusahaan khususnya pengembang dapat mengetahui kualitas *Software Accounting* secara kuantitatif
 - Pengembang dapat memperbaiki code Software Accounting XYZ ini sesuai rekomendasi penulis yang didapat berdasarkan Design Pattern.
-
- Keterkaitan tugas akhir ini dengan perkuliahan yang telah dipelajari oleh penulis adalah Manajemen Kualitas Teknologi Informasi (MKTI). Penulisan ini termasuk dalam topik *Quality Management* pada peta jalan penulisan salah satu laboratorium yang ada di Jurusan Sistem Informasi yaitu Perencanaan dan Pengembangan Sistem Informasi (PPSI), dengan sub topik pengukuran kualitas perangkat lunak

LANDASAN TEORI || METRIK CHIDAMBER DAN KEMERER



LANDASAN TEORI || METRIK CHIDAMBER DAN KEMERER

				Metrik	Object Oriented Construct
				<i>Weighted Method per Class (WMC)</i>	<i>Class/Method</i>
				<i>Depth of Inheritance Tree (DIT)</i>	<i>Inheritance</i>
				<i>Number OF Children (NOC)</i>	<i>Inheritance</i>
				<i>Response For a Class (RFC)</i>	<i>Class/Message</i>
				<i>Coupling Between Object Classes (CBO)</i>	<i>Coupling</i>
				<i>Lack of Cohesion Method (LCOM)</i>	<i>Class/Cohesion</i>
Metrik Chidamber dan Kemerer	Hijau	Kuning	Merah		
WMC	$1 \leq x < 50$	$50 \leq x \leq 150$	$150 < x$		
DIT	$x < 5$	$5 \leq x < 8$	$8 \leq x$		
NOC	$x < 4$	$4 \leq x < 8$	$8 \leq x$		
CBO	$x < 14$	$14 \leq x \leq 150$	$150 < x$		
RFC	$x < 100$	$100 \leq x \leq 1000$	$1000 < x$		

(Sumber: Buku Linda H Rosenberg, Applying and Interpreting Object Oriented Metrics, 1998)

LANDASAN TEORI || ISO/IEC 9126-1

Quality
model

Fungsionalitas (Functionability)

Kehandalan (Reliability)

Kebergunaan (Usability)

Understandability

Efisiensi (Efficiency)

Keterpeliharaan (Maintainability)

Portabilitas (Portability)

Replaceability

LANDASAN TEORI || ISO/IEC 9126-1

Metric	<i>Desirable Value</i>	<i>Efficiency</i>	<i>Understandability</i>	<i>Maintainability</i>	<i>Replaceability</i>
WMC	↓		↑	↑	↑
DIT	↓	↑	↑	↑	↑
NOC	↓	↑		↑	↑
CBO	↓	↑			↑
RFC	↓		↑	↑	
LCOM	↓	↑		↑	↑

Properti Kualitas Software	Parameter Metric
<i>Efficiency</i>	LCOM, CBO, DIT, NOC
<i>Understandability</i>	WMC, RFC, DIT
<i>Reusability/ Replaceability</i>	WMC, LCOM, CBO, DIT, NOC
<i>Maintainability</i>	WMC, RFC, DIT, NOC.

(Sumber: *Buku Linda H Rosenberg, Applying and Interpreting Object Oriented Metrics*, 1998)

LANDASAN TEORI || DESIGN PATTREN

Design pattern secara singkat adalah pola desain perangkat lunak yang digunakan untuk memecahkan masalah tertentu yang selalu berulang-ulang sebelumnya.

Refactoring merupakan proses memperbaiki desain perangkat lunak yang sudah ada tanpa mengubah external behavior perangkat lunak tersebut.

Design Pattern adalah pedoman, sedangkan Refactoring adalah cara untuk memperbaiki code.


Refactoring akan dilakukan pada bagian kode program yang sudah teridentifikasi mengandung bad smells. (Yosef.2011)

LANDASAN TEORI || CLASS DIAGRAM

<i>Source</i>	<i>Dependency Kind</i>	<i>Target</i>	<i>Deskripsi</i>
<i>class/interface</i>	<i>extends</i>	<i>class/interface</i>	<i>class/interface extends</i> (perluasan) terhadap <i>class/interface/</i> yang lain.
<i>class/interface</i>	<i>contains</i>	<i>class/interface</i>	<i>class/interface</i> pada <i>source</i> berisi sebuah <i>class/interface</i> yang lain.
<i>class/interface</i>	<i>contains</i>	<i>field</i>	<i>class/interface/</i> pada <i>source</i> berisi <i>field</i> dari <i>class</i> yang lain.
<i>method</i>	<i>returns</i>	<i>class/interface</i>	<i>method</i> pada <i>source</i> mengembalikan sebuah nilai berdasarkan <i>class/interface</i> yang lain.
<i>method</i>	<i>has param</i>	<i>class/interface</i>	<i>method</i> pada <i>source</i> mendeklarasikan sebuah parameter berdasarkan <i>class/interface</i> yang lain.
<i>method</i>	<i>throws</i>	<i>class/interface</i>	<i>method</i> pada <i>source</i> mendeklarasikan <i>class/interface/</i> yang lain dalam mengirim/melempar clause-nya.
<i>method</i>	<i>calls</i>	<i>method s</i>	<i>method</i> pada <i>source</i> memanggil <i>method</i> pada <i>class</i> yang lain.
<i>method s</i>	<i>acceses</i>	<i>field</i>	<i>method</i> pada <i>source</i> mengakses <i>field</i> pada <i>class</i> yang lain.
<i>field</i>	<i>is of type</i>	<i>class/interface</i>	<i>field</i> pada <i>source</i> didasarkan pada <i>class/interface</i> yang lain.
<i>any</i>	<i>references</i>	<i>class/interface</i>	<i>remote</i> objek yang dipanggil oleh objek lain melalui <i>remote object refernces</i> .

<i>Relationship</i>	<i>Deskripsi</i>	<i>Dependency Kind</i>
<i>Associations</i>	<i>Class</i> yang memiliki atribut berupa <i>class</i> lain atau <i>class</i> yang harus mengetahui ekstensi <i>class</i> lain	<i>Contains</i>
		<i>Referencess</i>
		<i>Is Of Type</i>
<i>Dependencies</i>	Operasi suatu <i>class</i> yang menggunakan <i>class</i> lain atau <i>field</i> /atribut <i>class</i> lain	<i>Retruns</i>
		<i>Has Param</i>
		<i>Throws</i>
		<i>Calls</i>
		<i>Accesses</i>
<i>Nesting</i>	Sebuah <i>class</i> yang didefinisikan didalam <i>class (outer class)</i> lain.	
<i>Realization</i>	Sebuah <i>class</i> yang mengimplementasikan <i>interface</i>	<i>Implements (Interface)</i>
<i>Generalization</i>	<i>Class</i> yang menunjukkan hubungan warisan (<i>inheritance</i>) dengan <i>class</i> lainnya.	<i>Extendss</i>

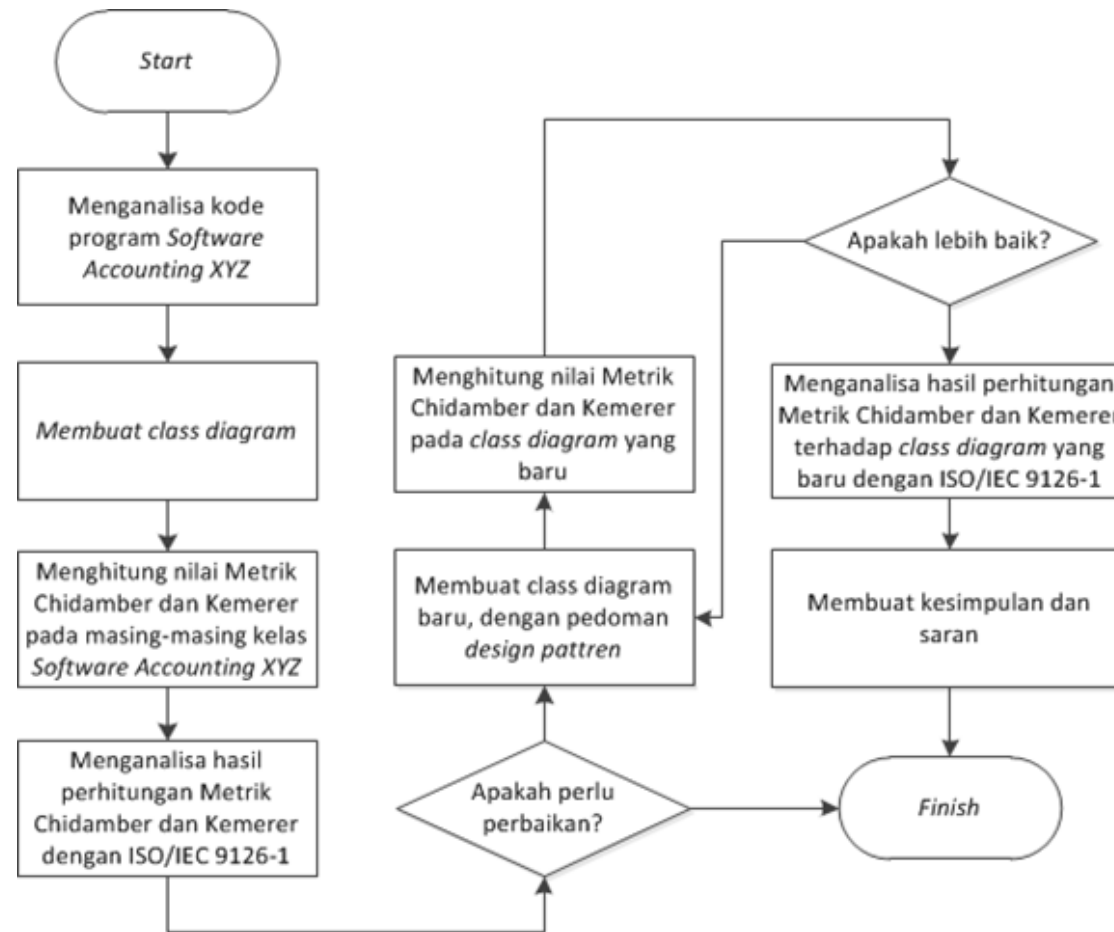
METODOLOGI PENELITIAN ||

INPUT	PROSES	OUTPUT
Tahap Perancangan		
Kode Program <i>Software Accounting XYZ</i>	Menganalisa kode program <i>Software Accounting XYZ</i>	<i>Code Dependency</i> dari masing-masing <i>package</i>
Kode Program <i>Software Accounting XYZ</i>	Membuat <i>class diagram</i>	<i>Class diagram</i> dari kode program <i>Software Accounting XYZ</i>
		
Tahap Implementasi		
<i>Code Dependency</i> dan <i>class diagram</i> dari <i>Software Accounting XYZ</i>	Menghitung nilai Metrik Chidamber dan Kemerer pada masing-masing <i>class Software Accounting XYZ</i>	Nilai WMC, DIT, NOC, CBO, RFC, dan LCOM

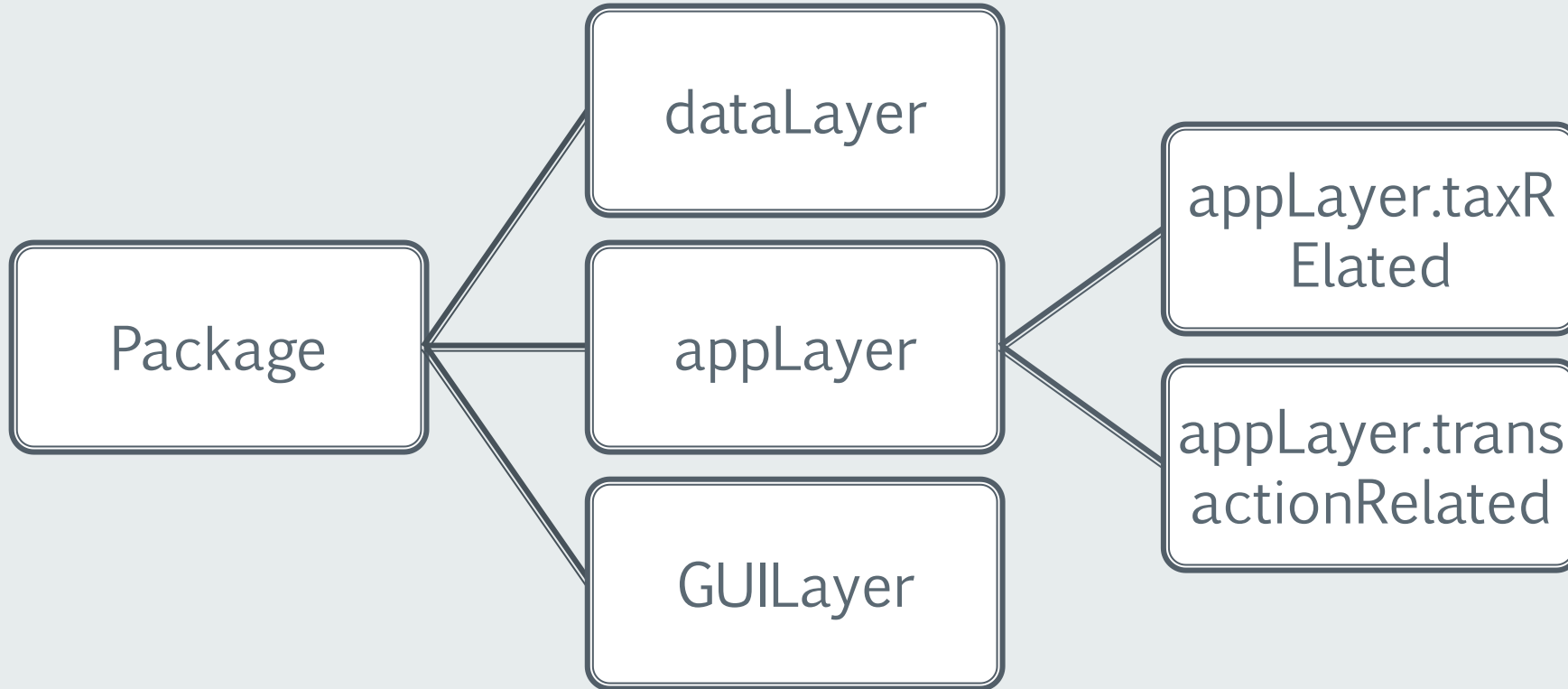
METODOLOGI PENELITIAN ||

Tahap Pembahasan Hasil		
Nilai WMC, DIT, NOC, CBO, RFC, LCOM dan ISO/IEC 9126-1	Menganalisa hasil perhitungan Metrik Chidamber dan Kemerer dengan ISO/IEC 9126-1	Kualitas <i>Software Accounting XYZ</i> berdasarkan ISO/IEC 9126-1
<i>Class Diagram</i> lama dan <i>Design Pattern</i>	Membuat <i>class diagram</i> baru	<i>Class diagram</i> baru
<i>Class diagram</i> baru	Menghitung nilai Metrik Chidamber dan Kemerer pada <i>class diagram</i> yang baru	Nilai WMC, DIT, NOC, CBO, RFC, dan LCOM yang baru
Nilai WMC, DIT, NOC, CBO, RFC, LCOM yang baru dan ISO/IEC 9126-1	Menganalisa hasil perhitungan Metrik Chidamber dan Kemerer yang baru dengan ISO/IEC 9126-1	Kualitas <i>Class Diagram</i> baru berdasarkan ISO/IEC 9126-1

METODOLOGI PENELITIAN ||



PERANCANGAN || GAMBARAN UMUM



152 Class

PERANCANGAN || CODE DEPENDENCY

Extend dependency

```
package appLayer.transactionRelated;
import java.sql.ResultSet;

public class cancelation extends transaction {

    public cancelation(transactions parent) {
        super(parent);
        // TODO Auto-generated constructor stub
    }
}
```

Class cancelation
extends (perluasan dari)
class transaction

Contains dependency

```
public class moneyTransfer {
    private HBCI hbcI=null;

    public moneyTransfer() {
        hbcI=HBCI.getInstance();
    }
}
```

Class moneyTransfer
berisi class HBCI yang ada
pada package dataLayer

Retruns dependency

```
package appLayer;
import java.text.DecimalFormat;

/*
 * This refers to the client of the application.
 * Customers are "contacts"
 */
public class client {
    public static transactions getTransactions() {
        if (transactionController==null) {
            transactionController=new transactions();
        }
        return transactionController;
    }
}
```

Method getTransactions
mengembalikan sebuah
atribut
transactionController
berdasarkan class
transactions

PERANCANGAN || CODE DEPENDENCY

Has param dependency

```
package appLayer;  
  
import ag.ion.noa.text.XInterfaceObjectSelection;  
import appLayer.transactionRelated.transaction;  
  
public class itemTableList {  
  
    private Shell shell = null;  
    private ITextDocument document = null;  
    private transaction theTransaction = null;  
    private placeholderManager placeholderManager = null;  
    private ITextField[] placeholderFields = null;
```

Class itemTableList
mendeklarasikan
parameter transaction
berdasarkan **class**
transactions

Throws dependency

```
package dataLayer;  
  
import java.io.BufferedReader;[]  
  
public class prepareDB {  
    public void createDB(Statement stmt, List listStatus) throws SQLException,  
        AccountNotFoundException {  
        String sql;
```

Class prepareDB mengirim
SQLException dengan mendeklarasikan
class AccountNotFoundException

Calls dependency

```
package appLayer;  
  
import java.util.Vector;[]  
  
class pageActivator implements appLayer.IItemListener {  
    private newTransactionSelectItemDetails parentPage=null;  
  
    public pageActivator(newTransactionSelectItemDetails parentPage) {  
        this.parentPage=parentPage;  
    }  
  
    public void onProductChange(product newProduct) {  
        // the wizard page can not be completed with the default product (<please  
        if (parentPage!=null) {  
            parentPage.checkPageComplete();
```

Methods onProductChange
memanggil **class**
checkPageComplete

PERANCANGAN || CODE DEPENDENCY

Accesses dependency

```
package applayer;
import org.kapott.hbci.OV.HBCIJob;

public class moneyTransfer {
    private HBCI hbci=null;

    public moneyTransfer() {
        hbci=HBCI.getInstance();
    }

    public void performMoneyTransfer(transaction theTransaction) {
        if (theTransaction.getCashFlow()==cashFlow.RECEIVE) {
            Konto own=new Konto(configs.getBankCode(), configs.getAccountCode());
            Konto other=new Konto(theTransaction.getRecipient().getBankCode(), theTransaction.
```

Methods
performMoneyTransfer
mengakses **fields** **RECEIVE**
pada **class** **transaction**.

Is of type dependency

```
package applayer;
import org.kapott.hbci.OV.HBCIJob;

public class moneyTransfer {
    private HBCI hbci=null;

    public moneyTransfer() {
        hbci=HBCI.getInstance();
    }
}
```

Field **hbci** didasarkan pada
class **HBCI**.

References dependency

```
import applayer.item;

public class fileImporters {
    public static void importKTimeTracker(String importFileName, StatuslineManager
        File importFile = new File(importFileName);
        if (!importFile.exists()) {
            item i=inv.addItem();
            i.setDescription(taskName);
            i.setQuantity(Double.valueOf(elements[elements.length-1]));
        }
    }
}
```

Method **fileImporters**
meremote dengan
memanggil **class** **item**.

Deatail pada buku hal 64 sampai 215

PERANCANGAN || CLASS DIAGRAM

<i>Source</i>		<i>Dependency kind</i>	<i>Target</i>		<i>Relasi</i>
<i>Package</i>	<i>Class</i>		<i>Package</i>	<i>Class</i>	
appLayer.taxRelated	IRSoffices	<i>contains</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
appLayer.taxRelated	IRSoffices	<i>calls</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
appLayer.taxRelated	IRSoffices	<i>references</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
appLayer.taxRelated	IRSoffices	<i>returns</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
appLayer.taxRelated	IRSoffices	<i>is of type</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
appLayer.taxRelated	state	<i>has param</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
appLayer.t	state	<i>calls</i>	appLayer.tax	IRSoffice	<i>Dependencies</i>

PERANCANGAN || CLASS DIAGRAM

	<i>Source</i>		<i>Dependency kind</i>	<i>Target</i>		<i>Relasi</i>
	<i>Package</i>	<i>Class</i>		<i>Package</i>	<i>Class</i>	
259	appLayer.taxRelated	states	<i>calls</i>	appLayer.taxRelated	state	<i>Dependencies</i>
260	appLayer.taxRelated	states	<i>references</i>	appLayer.taxRelated	state	<i>Associations</i>
261	appLayer.taxRelated	states	<i>returns</i>	appLayer.taxRelated	state	<i>Dependencies</i>
262	appLayer.taxRelated	tax	<i>contains</i>	appLayer.taxRelated	taxList	<i>Associations</i>
263	appLayer.taxRelated	tax	<i>has param</i>	appLayer.taxRelated	taxList	<i>Associations</i>
264	appLayer.taxRelated	tax	<i>calls</i>	appLayer.taxRelated	taxList	<i>Dependencies</i>
265	appLayer.taxRelated	tax	<i>is of type</i>	appLayer.taxRelated	taxList	<i>Associations</i>
266	appLayer.taxRelated	tax	<i>calls</i>	appLayer.taxRelated	taxList	<i>Dependencies</i>
267	appLayer.taxRelated	taxList	<i>returns</i>	appLayer.taxRelated	tax	<i>Dependencies</i>
268	appLayer.taxRelated	taxList	<i>calls</i>	appLayer.taxRelated	tax	<i>Dependencies</i>
269	appLayer.taxRelated	taxList	<i>references</i>	appLayer.taxRelated	tax	<i>Associations</i>

PERANCANGAN || CLASS DIAGRAM

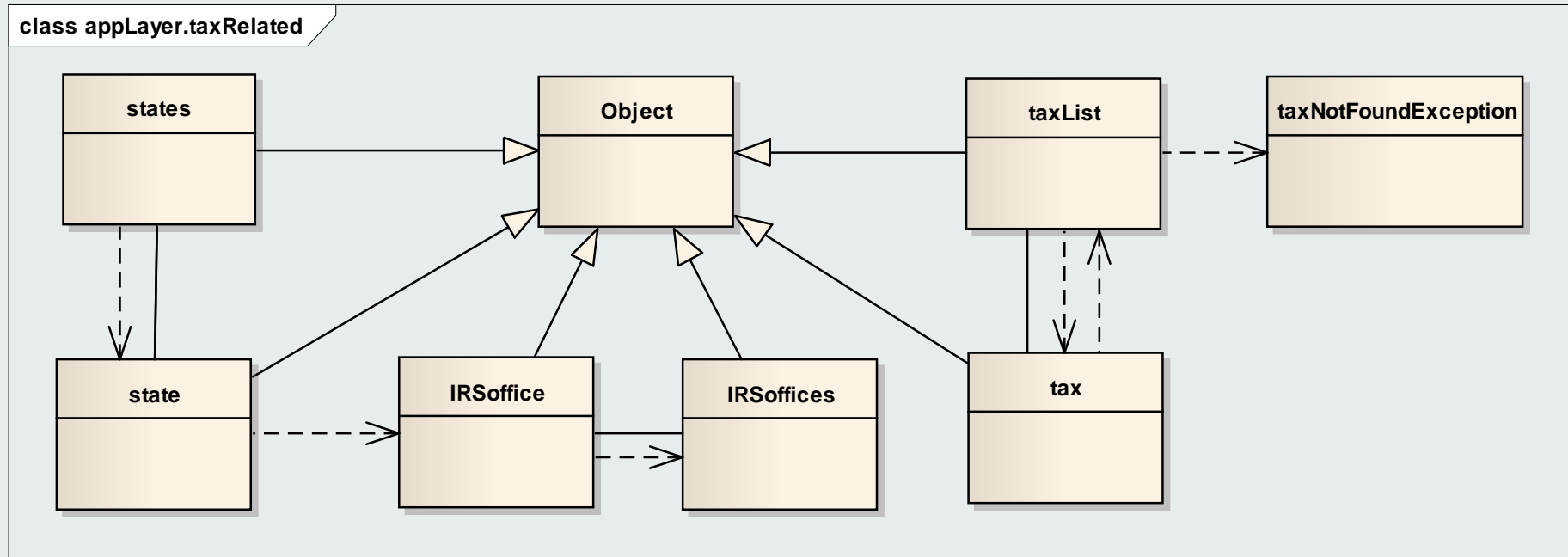
Source			Dependency kind	Target		Relasi
Package	Class			Package	Class	
270	appLayer.taxRelated	taxList	calls	appLayer.taxRelated	tax	Dependencies
271	appLayer.taxRelated	taxList	references	appLayer.taxRelated	tax	Associations
272	appLayer.taxRelated	taxList	returns	appLayer.taxRelated	tax	Dependencies
273	appLayer.taxRelated	taxList	has param	appLayer.taxRelated	tax	Dependencies
274	appLayer.taxRelated	taxList	throws	appLayer.taxRelated	taxNotFound Exception	Dependencies
275	appLayer.taxRelated	taxList	calls	appLayer.taxRelated	taxNotFound Exception	Dependencies

PERANCANGAN || CLASS DIAGRAM

252	appLayer.taxRelated	IRSoffices	<i>contains</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
253	appLayer.taxRelated	IRSoffices	<i>calls</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
254	appLayer.taxRelated	IRSoffices	<i>references</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
255	appLayer.taxRelated	IRSoffices	<i>returns</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
256	appLayer.taxRelated	IRSoffices	<i>is of type</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
257	appLayer.taxRelated	state	<i>has param</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
258	appLayer.t	state	<i>calls</i>	appLayer.tax	IRSoffice	<i>Dependencies</i>

Deatail pada buku hal 216 sampai 276

PERANCANGAN || CLASS DIAGRAM



Deatail pada buku hal 277 sampai 285

PERANCANGAN || CLASS DIAGRAM



Deatail pada lampiran A

[illegible]

Metrik NOC adalah jumlah anak *class*

NOC = 6

IMPLEMENTASI || METRIK WMC

Metrik Weighted Method per Class adalah jumlah semua method dan jumlah decision points (if/for/while) pada suatu class

WMC = 7

```
public class bookingContentProvider implements IStructuredContentProvider, Serializable {  
    /**  
    */  
    private static final long serialVersionUID = 1L;  
    Vector<entry> entries=new Vector<entry>();  
  
    public void addEntry(entry e) {  
        entries.add(e);  
        e.bindContentProvider(this);  
    }  
  
    public void removeEntry(entry e) {  
        entries.remove(e);  
    }  
  
    public void removeAllEntries() {  
        entries.removeAllElements();  
    }  
  
    public Object[] getElements(Object arg0) {  
        return entries.toArray();  
    }  
  
    public void dispose() {  
        // TODO Auto-generated method stub  
    }  
  
    public void inputChanged(Viewer arg0, Object arg1, Object arg2) {  
        // TODO Auto-generated method stub  
    }  
}
```

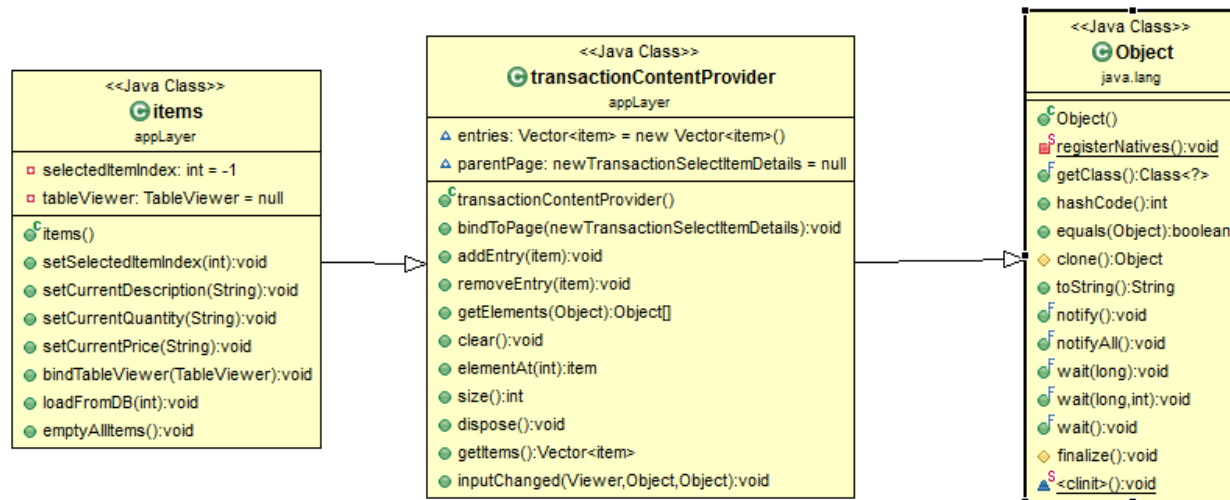
Class bookingContentProvider

IMPLEMENTASI || METRIK DIT

```
public class items extends transactionContentProvider {  
    private int selectedIndex = -1;  
    private TableViewer tableViewer=null;  
  
    public void setSelectedItemIndex(int index) {  
        selectedIndex=index;  
    }  
}
```

Class items

DIT = 2



IMPLEMENTASI || METRIK CBO

Package appLayer.transactionRElated mempunyai method dari class creditmemo yang memanggil class pada package appLayer dan appLayer.taxRelated, lebih detailnya dapat dilihat pada tabel code dependency. Class tersebut coupled dengan class account, class transactions, class AccountNotFoundException, class accountException, class taxList dan class entry. Sehingga class appLayer.transactionRelated.creditmemo mempunyai nilai CBO 6.

creditmemo	getTodoItems()	calls	account	getID()
creditmemo	creditmemo(transactions)	calls	accountsList	getRevenuesAccount()
creditmemo	creditmemo(transactions)	calls	accountsList	getBankAccount()
creditmemo	book()	calls	taxList	taxList()
creditmemo	book()	calls	taxList	getTaxesFromDatabase()
creditmemo	book()	calls	taxList	getDefault()
creditmemo	book()	calls	entry	entry(Date, String, Double, account, account, tax, String, String)

IMPLEMENTASI || METRIK RFC

Pada *class* *starmoneyImporter* terdapat dua *method* yaitu *starmoneyImporter* dan *method* *run*. Pada tabel *code dependency* dijelaskan bahwa *class* *starmoneyImporter* *calls method* *getClient()*,*getConnection*, *insert*, *invalidImportFormatException*, dan *parseCSVLineSemicolon*. Terdapat lima *method* yang dipanggil dan dua *method class* tersebut. Sehingga nilai RFC *class* *starmoneyImporter* adalah tujuh.

starmoneyImporter	run(IProgressMonitor)	calls	DB	getConnection()
starmoneyImporter	run(IProgressMonitor)	calls	DB	insert(String, HashMap<String, String>)
starmoneyImporter	run(IProgressMonitor)	calls	invalidImportFormatException	invalidImportFormatException()
starmoneyImporter	run(IProgressMonitor)	references	invalidImportFormatException	
starmoneyImporter	run(IProgressMonitor)	calls	fileUtils	parseCSVLineSemicolon(String)
starmoneyImporter	run(IProgressMonitor)	calls	client	getClient()

IMPLEMENTASI || METRIK LCOM

if $p > q$ *then* $LCOM = p - q$
else $LCOM = 0$

$p=5$ dan $q=1$, $LCOM = 4$

```
private TableView tableViewer;  
public bookingCellModifier(TableView tableViewer) {  
    this.tableViewer=tableViewer;  
public void modify(Object element, String property, Object value) {  
    tableViewer.refresh();  
}  
public boolean canModify(Object element, String property) {  
    return true;  
}  
public Object getValue(Object element, String property) {  
    return ((entry)element).getColumn(property);  
}
```

Methods

HASIL & PEMBAHASAN || ANALISA

No.	Metrik	Nilai
1	WMC	10,28
2	DIT	1,1
3	NOC	0,12
4	CBO	7,52
5	RFC	11,58
6	LCOM	20,2

Secara keseluruhan nilai Metrik Chidamber dan Kemerer sudah bagus

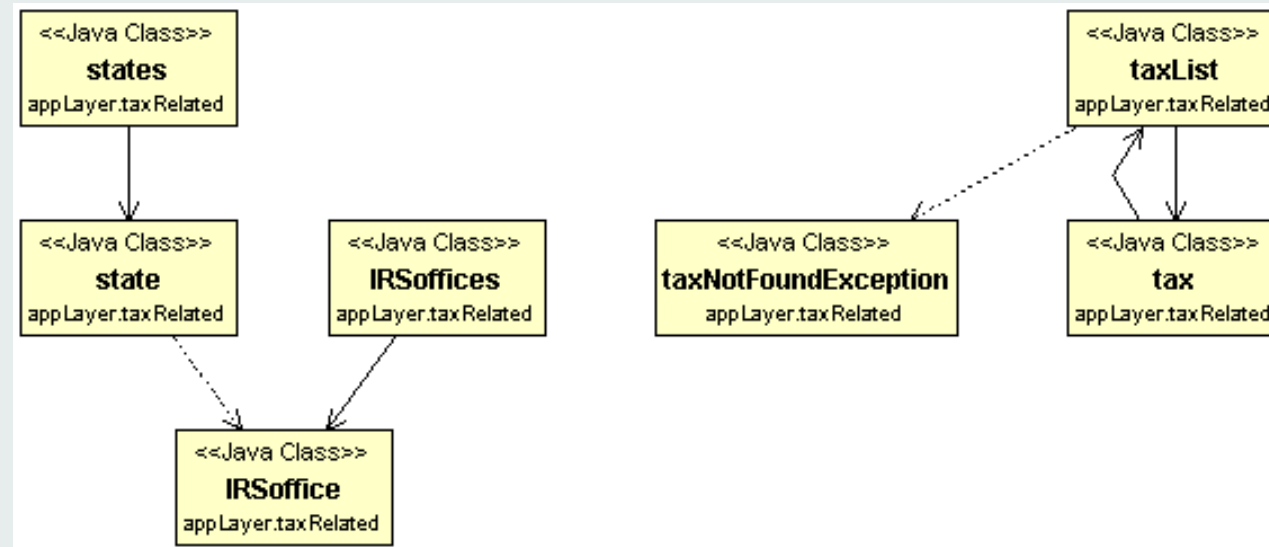
Nama Package	Class	Metrik	Nilai	Kategori
appLayer	accountsList	WMC	69	Medium
appLayer	configs	WMC	222	Bad
appLayer	appLayer.entry	WMC	132	Bad
dataLayer	dataLayer.DB	CBO	40	Medium
appLayer	appLayer.Messages	CBO	18	Medium
appLayer	account	CBO	15	Medium
appLayer	accountsList	CBO	19	Medium
appLayer	client	CBO	56	Medium
appLayer	configs	CBO	41	Medium
appLayer	entry	CBO	24	Medium
appLayer.taxRelated	taxList	CBO	16	Medium
appLayer.transactionRelated	transaction	CBO	26	Medium
appLayer.transactionRelated	transactions	CBO	18	Medium
GUILayer	Messages	CBO	40	Medium
GUILayer	balanceWindow	CBO	15	Medium
GUILayer	todoWindow	CBO	14	Medium
appLayer	entry	RFC	112	Medium

HASIL & PEMBAHASAN || CLASS DIAGRAM BARU

No	Jenis <i>Bad Smell</i>	Diskripsi	Solusi
1	<i>Duplicated Code</i>	Terdapat <i>code</i> yang sama dan berulang di lokasi yang berbeda.	Menghapus atau diimplementasikan sebagai satu <i>method</i> atau <i>function</i> tunggal yang akan dipanggil saat diperlukan
2	<i>Long Method</i>	<i>Method</i> yang ditulis terlalu panjang	Diubah dan dikelompokkan menjadi <i>method</i> yang lebih pendek.
3	<i>Large Class</i>	<i>Class</i> yang besar (terlalu banyak <i>method</i>) sehingga sulit di dibaca, di mengerti, dan di <i>maintenance</i> .	Dikelompokkan menjadi beberapa <i>class</i> atau dijadikan <i>partial class</i>
4	<i>Long Parameter List</i>	Parameter sebuah <i>method</i> terlalu banyak sehingga sulit di mengerti	menggunakan tipe objek sebagai parameter

5	<i>Feature Envy</i>	Sebuah <i>method</i> membutuhkan banyak informasi dari <i>class</i> lain daripada <i>class</i> nya sendiri	Mengelompokkan <i>method</i> yang sesuai
6	<i>Data Clumping</i>	Kelompok data yang sama (<i>field</i> dalam <i>class</i> , parameter dalam <i>method</i>) terulang kembali di beberapa tempat dalam sebuah program.	Mengganti dengan enkapsulasi semua data dalam satu objek
7	<i>Switch Statement</i>	Switch sebagai pengganti “if”	Menggunakan <i>Polymorphism</i>

HASIL & PEMBAHASAN || PERHITUNGAN METRIK CK



Deatail terdapat pada halaman336 sampai 342

HASIL & PEMBAHASAN || PERHITUNGAN METRIK CK

No.	Metrik	Nilai	Keterangan
1	WMC	8,83	Turun
2	DIT	1,08	Turun
3	NOC	0,09	Turun
4	CBO	6,27	Turun
5	RFC	10,46	Turun
6	LCOM	32,35	Naik

Secara keseluruhan nilai Metrik Chidamber dan Kemerer menjadi lebih baik

KESIMPULAN & SARAN

Class diagram baru yang diperbaiki berdasarkan *refactoring* tetap mempunyai 152 *class* namun, terdapat perubahan *method* sehingga tidak ada *class* yang *lazy* atau *class* yang kelebihan *method*.

Nilai Metrik Chidamber dan Kemerer lebih baik pada *class diagram* yang baru.

Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik WMC yang semula 10,28 menjadi 8,83; hal ini menunjukkan kenaikan atau perbaikan *understandability*, *maintainability*, dan *replaceability* sebesar 14,11% berdasarkan jumlah *method* atau kompleksitas setiap *class*.

Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik DIT yang semula 1,1 menjadi 1,08; hal ini menunjukkan kenaikan atau perbaikan *efficiency*, *understandability*, *maintainability*, dan *replaceability* sebesar 1,82% berdasarkan kedalaman sebuah *class*.

KESIMPULAN & SARAN

Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik NOC yang semula 0,12 menjadi 0,09; hal ini menunjukkan kenaikan atau perbaikan *efficiency*, *maintainability*, dan *replaceability* sebesar 25% berdasarkan jumlah *subclass (child class)* setiap *class*.

Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik CBO yang semula 7,52 menjadi 6,27; hal ini menunjukkan kenaikan atau perbaikan *efficiency* dan *replaceability* sebesar 16,62% berdasarkan jumlah *Coupling* setiap *class*.

Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik RFC yang semula 11,58 menjadi 10,46; hal ini menunjukkan kenaikan atau perbaikan *understandability*, *maintainability*, dan *replaceability* AA sebesar 9,67% berdasarkan jumlah *Message* setiap *class*.

KESIMPULAN & SARAN

Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik LCOM yang semula 20,2 menjadi 32,35, hal ini menunjukkan kenaikan atau perbaikan *efficiency*, *maintainability*, dan *replaceability* sebesar 60,15% berdasarkan jumlah *cohesion* setiap *class*.

Perlu adanya perbaikan dan pengembangan dalam penelitian selanjutnya. Tugas akhir ini tidak menyertakan implementasi dari rekomendasi yang diusulkan. Tugas akhir ini dapat menjadi referensi untuk mengukur dan membandingkan kualitas perangkat lunak yang mempunyai fungsi sama. Metrik yang digunakan dalam penelitian ini tidak mengukur semua karakteristik kualitas pada ISO/IEC 9126-1, perlu adanya perpaduan Metrik Chidamber dan Kemerer dengan metrik lainnya agar dapat mengukur semua karakteristik ISO/IEC 9126-1



TERIMA KASIH

-Genggamlah Impianmu Sampai Suatu Saat Mimpimu Terkabal-